The Need for Multiple AES Winners

By Brian Gladman, Worcester, United Kingdom

Background

The US National Institute of Standards and Technology (NIST) is currently leading a programme of work to define a replacement for the obsolescent Data Encryption Standard (DES) encryption algorithm. The current objective of this programme is to select a single algorithm – the Advanced Encryption Standard (AES) – in August 2000 to replace DES from then on.

The NIST AES programme is a truly remarkable effort to provide a next generation cryptographic standard. NIST has worked hard to ensure that this activity is international in character and this has ensured that many of the world's best cryptographers are involved.

This paper argues that the current objective of the AES effort is wrong in one respect. In order to provide protection against the possible future failure of a single algorithm, this paper argues that the objective of the AES effort should be to select three standard algorithms to replace DES rather than just one.

The Need for Multiple AES Winners

Many of the 15 AES candidate algorithms have been designed by the world's best academic cryptographers. The majority of these candidates are highly regarded for their security and there seems little doubt that when the finalists are selected later this year, all of them will be of very high quality and performance.

The essential difficulty created by selecting a single AES winning algorithm is that this will inevitably be used to protect a significant proportion of the world's critical data. If this algorithm is subsequently found to have a flaw, this will create a crisis since a large proportion of the world's critical data will suddenly be at risk. If, however, more than one AES algorithm is available, many applications will then be able to reduce the impact of this risk by offering a choice of algorithms.

Putting 'all our eggs in one basket' poses a well known security risk that all conservative security designers will seek to avoid. This risk cannot always be removed but in situations where this is not possible it is important to evaluate the consequences in order to be sure that the risks involved can sensibly be tolerated.

With a single AES winner, we will certainly be in a situation where the probability of algorithm failure is very small. At the same time, however, the extensive global use of this algorithm will put a significant proportion of the world's critical data at risk if the algorithm were to fail. The 'cost' of such a failure will hence be very large.

In an informal way we can judge the overall 'risk' as the product of an incredibly small probability of algorithm failure with the unimaginably large 'cost' if such a failure were to occur. We then need to know whether the result of this calculation is small enough to be tolerated or sufficiently large to require our attention.

Those who argue for one winner do so by suggesting that this risk is small. But they do not know this to be true – they are simply guessing that this is the case. In practice we have no idea what the probability of algorithm failure is and no idea, either, of the 'cost' of failure. In this situation, a conservative security designer will introduce measures to deal with this risk even though its magnitude is not known. Of course, this effort might later be seen as wasted if the risk does not materialise, but many will accept that this is a price worth paying in order to avoid a possible catastrophe. In other words, it is not sensible to take a 'calculated risk' unless we can actually calculate the magnitude of the risk we are taking.

In the AES case, we can't calculate the extent of this risk and this means that it is prudent to adopt a conservative approach. By selecting, for example, three winning algorithms rather than one, we can mitigate the risk of algorithm failure. Moreover, if we make this choice in an appropriate way [reference 1] we can further reduce these risks by reducing the chances that all our chosen algorithms will fail.

How many Winners?

If we choose too few winners we will fail to get the redundancy we need but interoperability and implementation costs will suffer if we choose too many. If we assume that the probability of algorithm failure (P_F) is the same for all algorithms, when we choose N winners the probability of all N failing will be $(P_F)^N$. By choosing two algorithms instead of one, we reduce the chances of catastrophic failure by a factor of P_F, whereas a choice of three reduces this by a factor of $(P_F)^2$.

The decision on how much algorithm redundancy to have is subjective. A non-conservative designer will go for none whereas someone who is ultra-conservative will want a large number of alternatives. In practice, however, P_F is already small and implementation cost and interoperability get worse as the number of algorithms increases and this suggests that the practical choice is either two or, at most, three winners.

Arguments Against Multiple Winners

A number of arguments are used to suggest that the selection of multiple AES winners is either unnecessary or is a bad idea.

Dilution of Cryptanalytic Efforts

It is often claimed that multiple algorithm selection will reduce the resources that the cryptanalytic community can deploy in finding AES algorithm weaknesses. Prior to algorithm selection this is not true since all algorithms have to be analysed in order to make a selection. After selection, it is true that such efforts will be diluted and this is one of the costs that have to be balanced against the benefits of multiple choice.

AES apart, such efforts are already devoted to a large number of algorithms, certainly a great many more than three. Once the AES selection process is completed, we are adding a small number of algorithms to the list that need to be analysed. The choice of three AES winners instead of just one would still allow each of these to be the subject of intense scrutiny, almost certainly a great deal more than many current algorithms receive.

Moreover, it is by no means certain that the concentration of the world's cryptanalytic resources on one algorithm rather than three would be more beneficial. With suitable algorithm choices [reference 1], it is possible that multiple AES winners will provide more overall benefit even in this respect.

Reduced Security

At the AES2 Conference, it is reported [reference 2] that Schneier argued against multiple choice by suggesting that if one bit of a key were used to select either of two AES algorithms, this would mean that half the time a poorer algorithm would be being used. This argument can, presumably, be developed in the following way.

If two winners are chosen, it is reasonable to assume that the choice of a single winner would be one of these two algorithms. If it is assumed that one of these is flawed, then in the case of a single winner there is a 50% chance that we will later find that our all our data is unprotected. If, instead, Schneier's proposal is adopted, we then find that 50% of our data is protected and 50% is not. Since for any data item in either case there is a 50% chance of failure, we have not gained anything by allowing multiple algorithms.

But this is not the issue – what matters is what we can do if a flaw is discovered. In the single choice situation we have nowhere to go when this happens – we are forced either to stop work or to continue without protection until a new algorithm can be introduced. If, however, we have alternative algorithms available, we can immediately switch to one of these and this means that we are able to continue working without disruption or risk.

This argument does not therefore undermine the case for multiple choice.

Increased Complexity and Cost

If applications have to implement multiple algorithms, their complexity and cost will be increased.

This cost is one that has to be traded against the benefits of multiple algorithm choice. However, current experience shows that many applications, for example, PGP and S/MIME, do provide multiple algorithms and this suggests that both providers and users see benefits in this. Most protocols are designed to cope with multiple (symmetric) algorithms and the additional cost of implementation in most software applications will not be significant when the number of alternatives is small.

It will be more difficult to implement multiple algorithms in smartcards but the fact that several algorithms are available does not mean that they all have to be implemented. Most smartcard applications operate in closed system contexts and this means that there are no interoperability complications in selecting one of the AES winners for such uses. Moreover, the fact that there is a choice may well provide a valuable degree of design freedom.

Although applications that use smartcards are likely to migrate towards open systems use, we can also expect their power to improve with time so that multiple algorithm implementation will no longer be a problem.

These considerations suggest that the impact of multiple algorithms on implementation cost and complexity can be effectively managed. The way in which cryptography is used in current software applications also supports the need for multiple algorithms.

Interoperability Problems

If too many choices are available, it is clear that a number of interoperability problems would result. However most modern protocols expect to make algorithm selections and most applications give users an algorithm choice. There is no reason to suggest, therefore, that the choice of two or three AES winners would lead to serious interoperability problems.

Multiple AES Algorithms Are Unnecessary

It is often argued that the selection of multiple AES algorithms is unnecessary because the redundancy objective, although valid, can be met either by using existing algorithms or by using 'AES losers'. These proposals suffer from a number of disadvantages.

First, if current algorithms were to be used, applications and protocols would have to cope with a great many different algorithms. It seems very likely that such an approach would create interoperability difficulties unless an effort was made to define a small subset of the currently available algorithms to be used for this purpose.

Second, current algorithms do not provide the standard block and key lengths that have been set for AES and this will mean that using non-AES algorithms as alternatives will not be as easy as using other AES candidates.

Third, using 'AES losers' without specifying which ones should be used could create interoperability problems. This would certainly be true if all fifteen were available but in practice it seems much more likely that the second round finalists would be used. This might involve four 'backup' algorithms – not an entirely unrealistic number – but probably too many in practice since some interoperability degradation might result. For this reason, it would be preferable to limit the alternatives to one or two of the losing AES candidates and this would be very little different from the selection of multiple AES winners.

A further, possible, problem with the informal use of 'AES losers' is that this may mean that some of the best candidates are no longer available for this purpose. This is because some AES design teams have only committed to make their algorithms freely available if they win. We might therefore find that the informal use of 'AES losers' as backups does not allow the choice of the best alternative algorithms. Of course, we don't know how the designers will react if the objective is changed to allow more than one winner but their stance can be determined if necessary.

Multiple Choice

When the author first proposed multiple AES winners, several commentators suggested that this would be good because it would allow the selection of algorithms which were individually the best in particular application domains. In other words, we could then select the best 'PC algorithm', the best 'smartcard algorithm', the best 'hardware algorithm' and so on.

The author considers this inadvisable because the objective of multiple choice is to achieve algorithm redundancy. If a particular algorithm excels in each individual application domain, it is then much more likely that this will be the only algorithm used here and this will remove the very redundancy that we are seeking.

If multiple AES winners are selected, it is therefore very important to ensure that all selected algorithms should provide good performance in all major application domains.

In the author's opinion, this is likely to be the biggest danger in the choice of multiple AES winners. Although unintended, we might find that AES winners are individually better suited to different domains to such an extent that redundancy is undermined. Particular care will be needed to avoid such an outcome.

Conclusions

The choice of a single AES winning algorithm involves a risk that the chosen winner will fail. The magnitude of this risk is unknown and this means that such an approach – putting all our eggs in one basket – is not prudent. It would hence be preferable to change the objective so that either two or three AES winners are selected. There will be costs in doing this but there is reason to believe that these are justified by the resulting benefit. In particular, experience with many current cryptographic applications shows that both providers and users see benefits in providing algorithm choice.

It is therefore recommended that either two or three AES winners should be selected. If such a change is agreed, it will be essential to ensure that all winners exhibit good performance across all major application domains.

Acknowledgement

I would like to thank John Myre of Sandia National Labs for pointing out an error in an earlier version of this paper.

References

1. "Future Resiliency: A Possible New AES Evaluation Criterion" by Don B. Johnson, Certicom, Second AES Conference.
2. 2. Live from the Second AES Conference, published on the sci.crypt mailing list by Dianelos Georgoudis.